The Dynamic Community of Interest and Its Realization in ZODIAC

Scott Alexander, Yuu-Heng Cheng, Brian Coan, Andrei Ghetie, Vikram Kaul, and Bruce Siegell, Telcordia Technologies

Steve Bellovin, Nicholas F. Maxemchuk, and Henning Schulzrinne, Columbia University

Stephen Schwab, SPARTA

Angelos Stavrou, George Mason University

Jonathan M. Smith, University of Pennsylvania

ABSTRACT

The ZODIAC project has been exploring a security first approach to networking based on a new idea, the dynamic community of interest, based on groups of users with a demonstrable need to know. ZODIAC uses the most challenging network setting (the mobile ad hoc network) as a target, since each node must incorporate functions of both hosts and routers. The realization of the DCoI is a work in progress, but initial implementation results have shown that DCoI concepts can be translated into working systems. The current system applies virtual machine containers, extensive use of cryptography and digital signatures, dispersity routing, DHT-based naming, and explicit rate control among other advanced techniques. Putting security to the forefront in the design has led to interesting consequences for naming, authorization, and connection setup. In particular, it has demanded a hierarchical structure for DCoIs that may initially appear somewhat alien to Internet users. Nonetheless, our implementation has illustrated that a highly available network that provides confidentiality and integrity can be constructed and made usable.

INTRODUCTION

A variety of proposals for redesign of the Internet have been made, often with end goals including better management [1], increased flexibility [2], and better support for mobility, among others. For the most part, these proposals start with the existing Internet and attempt to repair the aspect of the system that is of concern. For example, approaches like IPSec and virtual private networks (VPNs) deal with confidentiality and integrity, but not with availability. A somewhat different approach to a problem such as intrinsic assurance [3] is to do a true clean slate design of a networking architecture with security as a primary design goal, making other design choices to fulfill the remaining networking desiderata. This latter approach is the one we have taken in the ZODIAC project.

ZODIAC is a network architecture that puts security first and foremost, with security broken down into confidentiality, integrity, and availability (CIA). All three of these properties must be preserved for all applications of the system. For availability, adaptation and redundancy are the primary mechanisms that can be used. In a network architecture, routing over redundant paths can be used for highly available networking service [4]. For integrity and confidentiality, cryptography [5] and cryptography-based tools protect the links, but the routing nodes and hosts must use access control [6] to protect the CIA properties as well. Since nodes in a mobile ad hoc network (MANET) must serve as both routers and hosts, a unified solution for MANETs will work for hosts or routers as well.

DYNAMIC COMMUNITIES OF INTEREST

The basis of the ZODIAC design is a new distributed systems construct, the dynamic community of interest (DCoI). DCoIs provide a unit for which integrity, confidentiality, and availability are preserved. The DCoI corresponds directly to the security principle of need to know, applied to the elements of a distributed system. Information, whether file transfer or real time, is restricted by ZODIAC mechanisms to nodes that have a *need to know*, and can prove it through possession of appropriate credentials.

A DCoI is a dynamic group of networked nodes whose membership, application, and resources are regulated by its members as constrained by policy. Each of these elements is important to understanding the DCoI concept and necessary for the security provided by the DCoI. The narrow scope of each DCoI limits attack propagation, and supports confidentiality

This material is based on work supported by the Defense Advanced Research Projects Agency and Space and Naval Warfare Center, San Diego, under Contract no. N66001-08-C-2012.

 0163-6804/09/$25.00 © 2009 IEEE

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
OCT 2009		**00-00-2009 to 00-00-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
The Dynamic Community of Interest and Its Realization in ZODIAC	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Telcordia Technologies,One Telcordia Drive,Piscataway,NJ,08854-4151**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**8**	

and integrity through group keying. We look at each of these elements in turn.

It is important that DCoIs be dynamic so that they have the flexibility to handle all communications. If we were to allow any traffic to transit outside of the DCoI, it becomes possible for an attacker to avoid the defenses we put in place (e.g., quality of service [QoS] limitations and matching traffic against profiles). Moreover, if DCoIs were expensive to create or destroy, the temptation would arise to create a small number of them, each used for many purposes. This reduces their ability to enforce need to know separation, which in turn reduces the confidentiality and integrity of traffic within the system.

In a similar vein, we explicitly control the membership of the DCoI. Through mechanisms based on group services protocols, we require nodes to explicitly join each DCoI. This provides an opportunity to check the credentials of the node and enforce need to know. Nodes can also leave or be evicted when they no longer have a need to know or if they present a threat to the DCoI. These mechanisms provide confidentiality and integrity. Additionally, the membership policy can be used to balance availability. In particular, in the MANET environment adding additional well placed nodes may increase the system's ability to get packets to all nodes. This needs to be balanced against the danger that additional nodes present additional opportunities for insiders or other attackers to gain access to data. (We discuss this issue further later.)

Each DCoI supports a single application.[1] This decision represents one of the most radical changes from the Internet model. However, by constraining each DCoI to support a single application, we can more closely model what traffic should be seen within the DCoI. For example, if the DCoI is supporting voice over IP (VoIP) flows, we can expect to see our own control traffic as well as VoIP packets flowing within this DCoI. If we start to see frequent large packets attempting to flow within the DCoI, we can discard them without further processing. Within the model of signature-based intrusion detection, we are able to write a small number of rules for legal traffic rather than having to try to keep up with signatures for illegal traffic. This is key to transitioning from an allow-by-default system to a deny-by-default system. By denying an attacker's ability to access the system, we increase availability for legitimate users.

Resources are allocated to applications on a per-DCoI basis. Resources such as network bandwidth, CPU cycles, and memory are all allocated to the DCoI when it is created. These resources are then allocated to the application by the DCoI. This helps to ensure that if the application within a DCoI is successfully attacked, it cannot affect other DCoIs, and it only has a limited effect on well behaved nodes that are members of the DCoI. This also increases availability for legitimate users.

The DCoI is regulated by its members. In a MANET there is no centralized authority to enforce the rules and policies of the network. Moreover, since each node is both an end host and a router, malicious traffic may enter at any point. Therefore, in ZODIAC every DCoI node applies a broad set of protections at each network hop, enforcing DCoI policies across all aspects of networking, from rate control to content filtering. This is illustrated in the lower part of Fig. 1. This approach to assurance is in stark contrast to existing networks, which require blind (and uncontrollable) trust relationships between *red* applications and *black* network elements, and which permit propagation of attacks through multihop tunnels. The upper part of the figure illustrates this traditional approach. ZODIAC's comprehensive per-hop protections prevent such propagation while allowing DCoI-specific policies to control a more-complete range of network functions. Thus, every member of the ZODIAC network is authorized and expected to enforce the rules and policies of the ZODIAC system in order to protect itself, the network, and the DCoI. Since the rules and policies are designed to implement confidentiality, integrity, and availability, enforcement by each node increases each of these properties in the system.

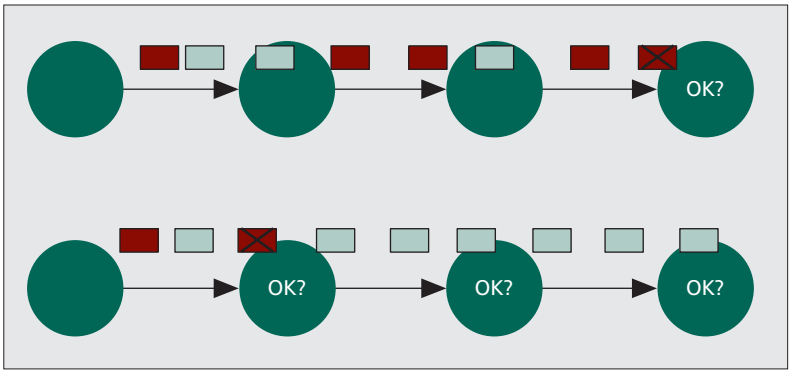

Figure 1. *Hop-by-hop enforcement.*

Finally, we use policy to provide the flexibility required for a deployed network. While there are basic ZODIAC rules that cannot be overridden by policy (e.g., the requirement that all traffic be encrypted), elements such as the resources allocated to a particular application or the list of members of the DCoI can only be selected at mission planning time or during the life of the DCoI. Our policy system is deny by default, so permission has to be explicitly given. Additionally, most policies are in effect only within the DCoI, allowing faster and more effective deconfliction across a much larger set of policies than traditional networks that apply and enforce common policies network-wide. The security properties improved by policy depend on the policies written. All three principles can be improved by well designed policies.

Figure 2 illustrates the architecture of a DCoI within a host. The dotted lines represent the boundaries of the DCoI containers. Within each of the containers, an instance of the services shown are running. We use SELinux and container protections to restrict egress of data to that path to the infrastructure network interface module. The thicker line along the left side of the container shows the data path as data moves to and from applications. The thinner lines with arrows show control paths. We do not explicitly show the encryption points (which would be after the QoS signaler for the transport parts of the packet and after the QoS forwarder for the

[1] *This brings up the obvious question, what is an application? We have intuitions, but do not have an answer yet. Application boundaries should be chosen to minimize the expensive operation of moving data between DCoIs, but prevent the possibility of data being shared inappropriately. We expect to develop a bright line answer after further experience with our system.*

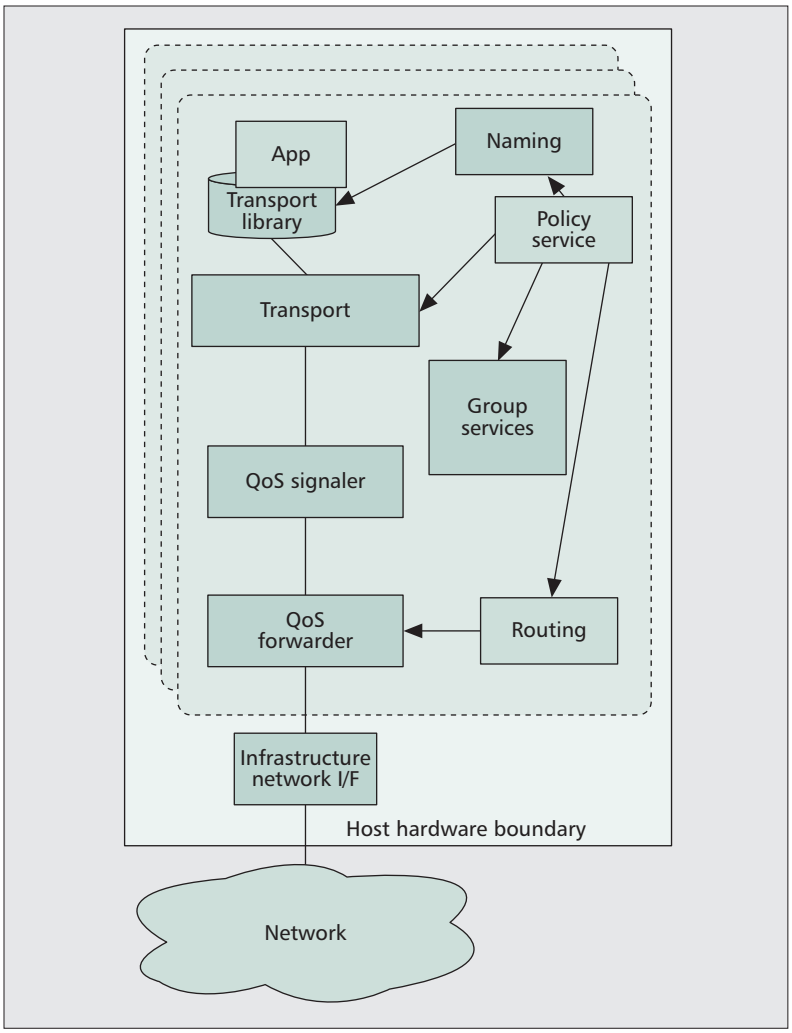

Figure 2. *The DCoI architecture.*

routing parts of the packet). Group services provide control in those processes as well as in the initial creation of the container.

Overall, the result is that ZODIAC provides confidentiality, integrity, and availability because it allows fine-grained control of network usage while limiting the scope of potential attacks. An attack must act like the application that is allowed in the DCoI, or its traffic will be discarded with very little processing. This typically requires that the attack use the same port numbers, and packet sizes and packet rates that would be appropriate to the application. Within those constraints and knowing the application, it becomes much easier to examine traffic for legal messages to further limit the behavior of the attack. Each of these defenses requires relatively little processing on the part of the defender while further constraining the scope of the attack. The general principle applied here is that more awareness of legitimate application behavior creates greater constraints on illegitimate behavior.

The DCoI cryptographically protects all communications while allowing hop-by-hop detection of malicious traffic, preventing attack propagation. Authorized traffic within the DCoI receives the benefits of multipath routing and resource allocation (described below) to improve availability.

In order to implement DCoIs, we have a set of subsystems to provide critical network services. These are group and cryptographic services (GCS), routing, naming, policy, infrastructure, transport, and QoS. Additionally, we match the guarantees in the network on the host through our host security subsystem.

GCS manages the membership of DCoIs and handles keying material used to protect the confidentiality of packets in the network. Routing is used to move packets through the network in a way that is consistent with the DCoI architecture. Naming is a secure replacement for DNS. Policy is responsible for the dissemination and interpretation of the policies used to control the ZODIAC system. Infrastructure consists of the components necessary to move data within the host in a secure manner. In particular, it prevents exfiltration of data between DCoIs within the host. The ZODIAC transport is based on the Internet transport protocols, but is designed to work with the ZODIAC routing system and also to perform better in a MANET environment. Our QoS subsystem is designed primarily to constrain the resources available to an attacker. Finally, our host security subsystem is designed using SELinux and containers to protect DCoIs from each other within a node.

Figure 3 shows a simple illustration of two DCoIs in a MANET. Each of the vehicles participates in at least one DCoI. The left DCoI is running a VoIP application to allow voice communications between the member vehicles. The right DCoI similarly provides a chat application. Note that there is one vehicle that is in both DCoIs. However, our host security subsystem provides for strict limits on the passing of data between the two DCoIs to enforce need to know. Also notice that multiple paths are shown for routing data within the VoIP DCoI.

In the remainder of this article, we discuss the challenges of the MANET environment in the next section, followed by our routing subsystem, our transport, our GCS, and host security. Finally, we draw conclusions in the final section.

MANETs

MANETs are networks formed from cooperating sets of network nodes. Each node is a potential source and sink of traffic, but also serves the role of an intermediate node in paths for other network nodes. Since the nodes are mobile, the connectivity varies with time; thus, there are considerable dynamics to the topology with which nodes in the network are connected.

Since each node can serve as both a host and a router, security problems for both hosts and routers must be addressed, and thus provide an excellent environment for stressing the DCoI concept and validating any implementation of it. If the DCoI works for a MANET, it will work for any networked system.

Among the system-level challenges for DCoIs in MANETs are the control of resources, trust management, and control of information flows (e.g., membership information) in the face of potentially high topology dynamics. Consider, for example, the need to provide a push-to-talk voice channel over a packet network with a changing

topology. Changes in path length can induce voice dropouts, and changes in resource availability may require significant reductions in quality. Changes in connectivity can demand dynamic rekeying of groups as members join and leave the DCoI.

The MANET environment is not solely characterized by negative attributes. One observation we have made is that broadcast communications channels may allow for rich connectivity, beyond that achievable with point-to-point wired networks. An example application of this potential for a high degree of topological connectivity is the great potential for multiple path routing, which can result in high reliability, better security, and graceful degradation of aggregate capacity between source and sink.

The role of hosts in the MANET is complex and, given the desired isolation of DCoIs in the network, requires technology to allocate resources on the ZODIAC node corresponding to the policy for the DCoI. This includes virtualization, with effects on transport and applications, which are directly bound to DCoIs. The DCoI also has associated keying information, and part of our ongoing research is the identification of a minimal trusted computing base (TCB) for ZODIAC nodes.

Multiple-Path Routing

The term *routing* has had its technical meaning distorted to mean, even for many network specialists, *what Internet routers do*. In fact, today's Internet routers pursue a particular solution to a problem that has many possible solutions. (See, for example, the solution used by AT&T for high network availability in [7]). That is the problem of taking a graph representing the connectivity among a set of nodes (determined by the presence of links interconnecting the nodes) and determining paths from sources to destinations constructed from sequences of link traversals. From an assurance standpoint, the current Internet routing paradigm has two glaring shortcomings: choice of a single best path and reliance on information from possibly subverted remote nodes.

Choosing only a single *best* path between source and destination fails to leverage the full connectivity graph that the network provides (i.e., the multiplicity of parallel paths) at any given time. This approach impedes assurance in several ways. It relies on route updates to sense and respond to link outages. These updates are often slow enough to cause application timeouts and disruptions in mission-critical communication. It also exacerbates traffic bottlenecks, degradation in QoS, and susceptibility to security threats (e.g., adversarial SIGINT collection and traffic analysis).

The ZODIAC routing system addresses this issue in two ways. It uses geographic routing to avoid trusting route updates from distant nodes. It uses dispersity routing to choose multiple paths through the network. We explain each of these in turn.

Our geographic routing approach divides the MANET into routing zones, each of which is approximately one transmission range across. Figure 4 illustrates a network that fits into nine routing zones, with the gridlines representing the routing zones.[2] These routing zones are fixed with respect to the surface of the earth with the grid being set at mission planning time. Each node writes its location into the Zodiac Naming System. A node with a reason to communicate with another node is able to retrieve this location, which is then used as the destination for the packet. Each intermediate hop is responsible for forwarding the packet toward the destination based on what neighbors it has. In particular, if a malicious node forwards a packet in a bad direction, the next non-malicious node will forward it correctly, limiting the effect the malicious node can have. As illustrated in the figure by the solid line, each packet traverses a path via nodes in adjacent routing zones. The source node at the upper left looks up the destination routing zone of the node at the lower right and puts this into the packet header. Within each routing zone, neighbor information allows the packet to be forwarded to the adjacent routing zone traveling toward the ultimate destination.

Similarly, since we do not send routing updates around the network, nodes are unable to send bad information about their connectivity. This avoids the possibility of wormholes, black holes, and related attacks. Malicious nodes can, however, still execute attacks such as discarding packets, traffic analysis, or attempting to break the cryptography (e.g., with stolen keys).

To limit the effect of the remaining hazards, we use dispersity routing, a technique for sending the data from a single flow across multiple

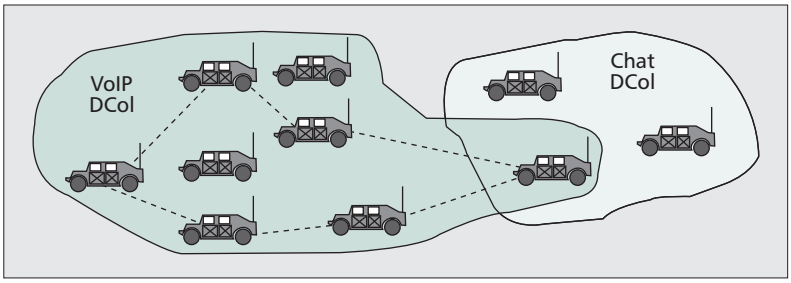

Figure 3. *Two DCoIs in the network.*

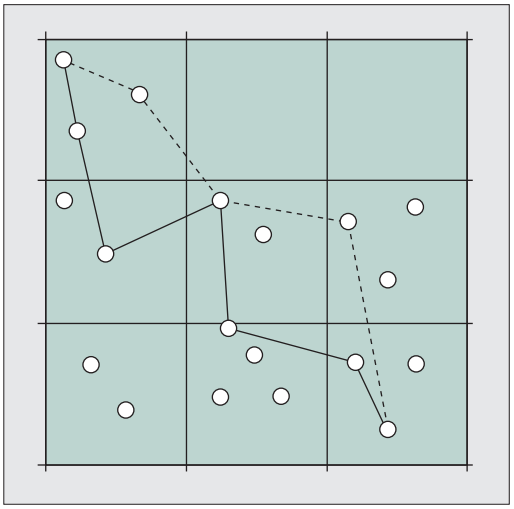

Figure 4. *Geographic and dispersity routing.*

[2] *It is more efficient to use hexagons for routing zones. We use squares in this discussion for simplicity.*

The DCoI structure is effective at ensuring that we avoid exfiltration and it eases the task of minimizing the effects of badly behaving applications on other network users.

paths through the network. Each source uses policy to determine the number of paths across the network desired for each flow. Packets are then routed in different directions through the network using geographic routing. Packets initially route toward an intermediate point in the network (a routing zone that may or may not actually contain any nodes) and then are redirected toward the target. This approach, of course, will cause packets to arrive out of order. Our transport layer expects this and provides reordering as necessary. Additionally, we are limited in cases where diverse paths do not exist because of the connectivity of the MANET. In cases where multiple paths do exist, we limit the influence any particular malicious node can have on the traffic that transits the node. Figure 4 illustrates two disperse paths with the solid and dotted lines. As shown, the paths may intersect at some points based on the location of actual nodes and the intermediate nodes selected by the source. Thus, dispersity is an element that raises the cost of a successful attack.

Additionally, we intend to implement monitoring of the paths. At first order, the endpoints do not care whether a path is underperforming because of a malicious node or because one of the nodes along the path has tenuous connectivity to its neighbor. In either case, it makes sense to either reduce the traffic across that path or discard the path altogether and potentially pick a new path as a replacement. As a secondary defense, we also intend to gather evidence of which node or set of nodes along the path is underperforming and to attempt to determine the cause.

The DCoI structure is effective at ensuring that we avoid exfiltration and eases the task of minimizing the effects of badly behaving applications on other network users. However, it can easily be the case that the members of a DCoI do not have sufficient connectivity to provide communications by themselves. In such a case, it is important to be able to move data across other nodes without making the data visible to those nodes.

In order to accomplish this goal, we allow data from one DCoI to be routed over another DCoI. We call the DCoI that originates the data the *application DCoI*. The DCoI that routes the data is called the *routing DCoI*. We constrain the relationship so that the routing DCoI must be an ancestor of the application DCoI. In particular, this ensures that all members of the application DCoI are also members of the routing DCoI.

In order to avoid the possibility of exfiltration of data, we use separate keys to encrypt the portions of the packet generated by the application DCoI and those generated by the routing DCoI. The result is that if an intermediate node is a member of the routing DCoI, but not the application DCoI, it can decrypt the routing header, which supplies sufficient information to forward the packet. (If the node is also a member of the application DCoI, policy determines whether it is sent up to the application DCoI and decrypted to allow for content filtering and other defensive measures. This use of policy allows us to trade increased CPU usage in the network with increased use of bandwidth for packets that will not be accepted by the end host.)

Through the use of routing DCoIs, we allow the mission planner to limit the number of nodes with access to the application data while still providing an appropriate level of connectivity. Moreover, since the routing DCoI's membership is controlled by policy (as is true of any DCoI), it is straightforward to implement policies such as requiring traffic to be carried by U.S. personnel only.

To provide a full suite of routing protocols, we have three additional interrelated types of routing beyond geographic routing. At the most basic level, we provide flooding. This flooding is limited to the routing DCoI and hence can be a rather efficient choice for multicast flows when the recipients are a large percentage of the nodes in the routing DCoI. Additionally, we use flooding as a bootstrap mechanism when we do not have sufficient information to use our other techniques.

Additionally, we have implemented Optimized Link State Routing (OLSR) in ZODIAC in order to have a basis of comparison. We have not attempted to secure the protocol itself. However, since each DCoI routes independently, a failure in one routing DCoI will not affect other routing DCoIs. Additionally, the other protections provided by the ZODIAC system reduce the ability of an attacker to spoof OLSR messages, eavesdrop, or launch host-based attacks on the OLSR process.

Finally, we have a tree-based multicast approach. This approach is designed so that data sources advertise the availability of a flow and do not reveal the identities of any consumers of the data. The trees are built dynamically within the network based on the connectivity of subscribers.

Unicast and Multicast Transport

ZODIAC transport is divided into four types, which are the cross product of reliable, unreliable, unicast, and multicast. Reliable unicast is based on IP QoS, standardized in the Telecommunications Industry Association (TIA) 1039 specification [8]. Unreliable multicast is largedly addressed by the routing system, and unreliable unicast resembles User Datagram Protocol (UDP) constrained by resource limits.

We have chosen TIA-1039 due to its measured performance for high bandwidth □ delay product networks in the face of high error rates. The multiple hops of a MANET are a source of delay, independent of bandwidth, and the wireless nodes are subject to the many errors inherent in radio frequency communications. TIA-1039 achieves its performance by transforming the discovery of available link capacity from multiple round-trip times to a single round-trip time. It does this by signaling rates explicitly as packets traverse nodes behaving as routers in the MANET. As a packet traverses nodes, the available rate is updated in its header, requiring considerable interaction with the security environment, since each node in the path must be able to read and write the packet headers.

The reliable unicast protocol is called ZODIAC Control Protocol or ZCP, and has been implemented under Linux. It is implemented as a user-level transport stack, and this decision to

prototype at the user level has proven valuable in design and debugging, as it has allowed for rapid update and redeployment. Performance enhancements have followed a traditional path, focusing on algorithms first, but with attention also paid to issues such as concurrency and copying.

GCS

ZODIAC GCS provides several key functions to the DCoI. It allows nodes to join or leave the DCoI and to be evicted. It also determines when rekey operations are necessary (based on both elapsed time and events such as evictions of nodes). It additionally provides the interface to the cryptographic services required by the ZODIAC system.

Particular challenges for GCS are dealing with the partitions that can commonly occur in a MANET environment and dealing with Byzantine faults. Our handling of both of these challenges are described in more detail in [9]. In brief, we use multiple group controllers configured to provide protection against a preselected number of malicious controllers. Since a node trying to join the DCoI contacts multiple group controllers, it can be certain that it gets a valid key for use in accessing DCoI resources. Moreover, the communications between the group controllers, and between the members and group controllers are designed so that the DCoI can merge state and continue to operate after a partition has healed.

This allows for dynamic membership with decentralized group controllers. GCS is also closely integrated with our policy system so that policies related to group membership are consistent with and deconflicted with other policy decisions in the network. In order to deal with the challenge of Byzantine faults, group controllers perform signing using key shares. Keying material is currently prepositioned; we plan to investigate dynamic key distribution as part of our future work. For additional details on the structure and design of GCS, see [9].

Within ZODIAC , we use keying material for a variety of operations, from checking the credentials of a node attempting to join a DCoI to encrypting and decrypting data in packets to be sent across the network. We have placed the functions that access this material in GCS. This has the advantage of minimizing the amount of code that needs to be validated as correctly handling keying material. It additionally ensures that it is straightforward to change algorithms or cryptographic implementations. Only the code in GCS is affected.

Host Management

The ZODIAC system is designed so that the protections afforded in the network extend into the host. This provides a true end-to-end solution rather than the more typical mismatched attempts to secure the network and the host independently. Moreover, we assume that the host will run applications that were not securely implemented.

This, in particular, provides a contrast with prior systems such as protected core networks (PCNs) [10]. PCNs and similar approaches protect the core to provide subnet-to-subnet protections. The subnets at the edges provide their own protection. Additionally, ZODIAC controls most control traffic to be sensitive as well. Thus, for example, routing messages within a DCoI are protected to the same level as user data. We believe that this allows ZODIAC to extend the level of protection provided by PCNs in environments where it is required.

We use a combination of virtual machine [11] and SELinux [12] protections to implement the DCoIs on the host. Each DCoI is represented on the host as a virtual machine container. This container is configured in such a way that it has only a small number of controlled paths to provide access to outside the container. These paths are controlled by our infrastructure subsystem. It controls the ability of processes within the container to send or receive external data, determines where that data may be sent, and ensures that encryption and decryption occur as required by the system design. In particular, it controls all access to the network interface. This ensures that no traffic can be sent or received unless it meets the requirement of the system design.

For example, as described previously, we describe how traffic from an application DCoI can be routed over an ancestor that provides greater connectivity. However, if the application DCoI protocol data unit (PDU) were given to the routing DCoI in the clear, information could be exfiltrated through the routing DCoI. Infrastructure therefore ensures that the application DCoI PDU is encrypted with the application DCoI key before the PDU is available to the routing DCoI. Information needed by the routing DCoI (e.g., the intended destination) is passed as metadata in a format known to infrastructure so that it can be verified against the system requirements and system policy.

We use SELinux to provide fine-grained security in places where the broad brush of the containers is not sufficient. For example, we use a UNIX domain socket to allow an infrastructure process running inside the container to talk to the infrastructure process running outside the container. To ensure that no race condition can exist on creating and connecting to that socket, SELinux is used. It is configured so that only the intended processes can use that socket. Additionally, SELinux is able to provide the protection of only allowing intended processes to run inside the container (thus avoiding attacks where code is downloaded and then executed). While it might be possible to provide our entire host side security solution through SELinux, the virtual machine container approach has the advantage of requiring much less configuration for the default parts of the system. Since our goal is to isolate elements inside the DCoI from anything outside the DCoI, containers give a simple design for this goal.

Conclusions

The ZODIAC system is still in its early stages. There is much work to be done. In this article we have discussed the architectural principles. As the implementation progresses, we expect to gain experience to allow us to answer some of

The reliable unicast protocol is called ZODIAC control protocol or ZCP, and has been implemented under Linux. It is implemented as a user-level transport stack, and this decision to prototype at user level has proven valuable in design and debugging, as it has allowed for rapid update and redeployment.

We also believe that the ZODIAC model is appropriate for wired networks. As with IP networks, a gateway would be required between the wired routing protocol and the wireless routing protocol and those protocols would tend to be different in nature.

the remaining questions. One of the most pressing of these asks how dynamic DCoIs can be. The answer, of course, will have to be parameterized by resource availability including CPU and bandwidth.

We have implemented a proof-of-concept gateway between a Zodiac MANET and a pair of IP subnets running over Ethernet using Mobile Ad Hoc Network Emulator (MANE) [13] as an emulator for the network. This shows that it is possible to have IP nodes and Zodiac nodes communicating, each with unmodified stacks. The next step in this work will be to determine which security properties we can maintain and how to provide adequate performance.

We also believe that the ZODIAC model is appropriate for wired networks. As with IP networks, a gateway would be required between the wired routing protocol and the wireless routing protocol, and those protocols would tend to be different in nature. We believe that the other components of our architecture are appropriate for a wired network in the sense of providing connectivity.

Of course, if one were to move outside the context of military communications, there are additional questions raised about the trade-off between increased security and the flexibility of the original ARPAnet model. As the ARPAnet grew into today's Internet, of course, much flexibility was lost due to the centralizing effects of Internet service providers and backbone providers. There is an ongoing policy discussion as to whether providers should be allowed to block certain applications and protocols, and lawsuits regarding whether one has rights to anonymity. ZODIAC is designed for those environments where security is valued more highly than anonymity.

A related question is the scalability of the ZODIAC architecture. While we have designed the architecture for scalability, our experience is that implementation and use provides the true test of scalability.

We have described the ZODIAC system, which provides an application-to-application model of security. Through the use of the DCoI, we have described how we enforce need-to-know and deny-by-default policies, while providing flexibility sufficient for military networking needs.

Our design builds on a foundation of three core principles that every ZODIAC node must follow:

- Police all authorized traffic according to DCoI-specific policies for QoS, resource allocation (e.g., bandwidth, computing resources), content filtering, and routing. Scope DCoIs narrowly to constrain the effects of any attack or failure within the DCoI.
- Comprehensive hop-by-hop enforcement within the DCoI: drop traffic that is not cryptographically authorized and protected, or that violates prenegotiated constraints.
- Protect the system against byzantine failures through a diversity of techniques, including dispersity routing, network coding, distributed naming, keying, and trust management services, plus QoS and transport protocols tailored to multipath routing.

These three principles represent a major departure from the practices of the conventional Internet. For example, applying a range of assurance mechanisms hop by hop, spanning several layers of the traditional protocol stack at each DCoI node, eliminates the need for trust relationships between applications and black-side network elements, and constrains attacks (including worms) to one-hop neighbors within the DCoI. The requirement for cryptographic signatures also provides for non-repudiation and accountability that the conventional Internet does not provide. In addition to limiting the scope of attacks, the second principle constrains the space of policies and other configuration parameters pertaining to each DCoI, facilitating safety measures against misconfiguration. The third principle provides several forms of redundancy to protect against byzantine or conventional failures, thereby increasing overall availability and integrity.

Acknowledgments

The authors would like to thank the ZODIAC team who are currently in the midst of implementing this architecture. Their contributions to the design and experience with the architecture will ultimately prove the basis for evaluating this system. We would also like to thank Timothy Gibson and Chris Ramming of DARPA as well as Pete Sholander, David Duggan and their independent test and evaluation team for their support and insights throughout the program. We also appreciate insights on PCNs provided by Timothy Gibson. Finally, we would like to thank the anonymous reviewers for their thoughtful comments, which have strengthened this article.

References

[1] D. D. Clark *et al.*, "A Knowledge Plane for the Internet," *Proc. SIGCOMM*, 2003.
[2] J. M. Smith and S. M. Nettles, "Active Networking: One View of the Past, Present, and Future," *IEEE Trans. Sys., Man, & Cybernetics, Part C: Apps. and Reviews*, vol. 32, no. 1, Feb. 2004, pp. 4–18.
[3] S. Alexander *et al.*, "Requirements and Architectures for Intrinsically Assurable Mobile Ad hoc Networks," *Proc. MILCOM*, Nov. 2008, pp. 1–10.
[4] N. F. Maxemchuk, "Dispersity Routing," *Proc. ICC*, vol. 41, June 1975, pp. 10–13.
[5] A. Keromytis, J. Ioannidis, and J. M. Smith, "Implementing IPSec," *Proc. IEEE GLOBECOM*, Nov. 1997, pp. 1948–52.
[6] B. Lampson, "Protection," Operating Sys. Review, vol. 8, no. 1, 1974, pp.18–24.
[7] G. R. Ash *et al.*, "Real-Time Network Routing in the AT&T Network: Improved Service Quality at Lower Cost," *Proc. GLOBECOM*, 1992, pp. 802–9.
[8] L. Roberts, "QoS Signaling for IP QoS Support," *Tech. Rep. TIA-1039*, TIAn, 2005.
[9] J. Kirsch and B. Coan, "Intrusion-Tolerant Group Management for Mobile Ad-Hoc Networks," Dependable Network Computing and Mobile Systems Wksp., accepted for publication, Sept. 2009. Preliminary version available as: J. Kirsch and B. Coan, "Instrusion-Tolerant Group Management for Mobile Ad-Hoc Networks," Tech. Rep. CNDS-2009-2, Johns Hopkins Univ., 2009.
[10] G. Hallingstad and S. Oudkerk, "Protected Core Networking: An Architectural Approach to Secure and Flexible Communications," *IEEE Commun. Mag.*, vol. 46, no. 11, Nov. 2008, pp. 35–41.
[11] P. Barham *et al.*, "Xen and the Art of Virtualization," *Proc. SOSP*, 2003.
[12] P. Loscocco and S. Smalley, "Meeting Critical Security Objectives with Security-Enhanced Linux," *Proc. Ottawa Linux Symp.*, 2001.

[13] Network and Communication Systems Branch, Mobile Ad Hoc Network Emulator (MANE); http://cs.itd.nrl.navy.mil/work/ mane/index.php

BIOGRAPHIES

SCOTT ALEXANDER (salex@research.telcordia.com) is a senior scientist at Telcordia and is the PI for the DARPA-funded Zodiac project. His research interests cover security and networking, particularly in challenging environments. He has worked on a variety of projects ranging from writing a command system for deep space spacecraft to redesigning network security. He holds a Ph.D. and M.S. in computer and information sciences from the University of Pennsylvania as well as a B.A. in computer science from Rice University.

STEVEN M. BELLOVIN is a professor of computer science at Columbia University, where he does research on networks, security, and especially why the two don't get along. He joined the faculty in 2005 after many years at Bell Labs and AT&T Labs Research, where he was an AT&T Fellow. He received a B.A. degree from Columbia University, and his M.S. and Ph.D. in computer science from the University of North Carolina at Chapel Hill. While a graduate student, he helped create Netnews; for this, he and the other perpetrators were given the 1995 Usenix Lifetime Achievement Award (The Flame). He is a member of the National Academy of Engineering and is serving on the Department of Homeland Security's Science and Technology Advisory Committee; he has also received the 2007 NIST/NSA National Computer Systems Security Award. He is the coauthor of *Firewalls and Internet Security: Repelling the Wily Hacker*, and holds a number patents on cryptographic and network protocols. He has served on many National Research Council study committees, including those on information systems trustworthiness, the privacy implications of authentication technologies, and cybersecurity research needs; he was also a member of the information technology subcommittee of an NRC study group on science vs. terrorism. He was a member of the Internet Architecture Board from 1996 to 2002; he was co-director of the Security Area of the IETF from 2002 through 2004.

YUU-HENG CHENG [M] is a senior research scientist in Applied Research at Telcordia Technologies. She received her B.S. and M.S. degrees in computer science and information engineering from National Chiao-Tung University in 2001. She was involved in policy-based network management, network security and compliance, device mobility management, and standards activities. Her primary research area is policy-based system architecture for distributed systems.

BRIAN COAN is director of the Distributed Computing group at Telcordia, where he has worked since 1978. He works on providing reliable networking and information services in adverse network environments, possibly caused by cyber attacks. He received a Ph.D. in computer science from MIT in the Theory of Distributed Systems Group in 1987. He holds a B.S.E. degree from Princeton University (1977) and an M.S. from Stanford (1979). He has twice served on the program committee of the ACM Conference on the Principles of Distributed Computing. He is a member of the ACM.

ANDREI GHETIE is a senior scientist at Telcordia. His research interests are in network QoS, policy, and the interactions between the two. He frequently moves between projects scheduled for deployment and fundamental research to keep a broad perspective. He has an M.S. in information networking from Carnegie Mellon University and a B.S. in computer systems engineering from Stanford University.

VIKRAM KAUL is a senior research scientist in the Wireless Systems and Networks Research group at Telcordia Technologies, where he currently does applied research on 3G radio network optimization, ad hoc multicast routing, and tactical application data filtration. He has previously worked on several topics including intrusion detection, ad hoc routing, QoS, UWB MAC protocols, distributed provisioning, and network management. He received a B.E. (Hons) degree from BITS Pilani, India, in 1997 and an M.S. from WINLAB, Rutgers University in 2000.

NICHOLAS F. MAXEMCHUK [F'89] received a B.S.E.E. degree from the City College of New York, and M.S.E.E. and Ph.D. degrees from the University of Pennsylvania. For the past eight years he has been a professor in the Electrical Engineering Department at Columbia University, New York. He has a dual appointment as a chief researcher in IMDEA Networks, Madrid, Spain. Prior to joining Columbia he spent 25 years at Bell Labs and AT&T Labs as a member of technical staff, department head, and technical leader. Prior to joining Bell Labs, he spent eight years at the RCA David Sarnoff Research Center as a member of technical staff. He has been Editor-in-Chief of *IEEE Journal on Selected Areas in Communications*, an editor for *IEEE Transactions on Communications* and *Journal of ACM*, was on the founding committee of *IEEE/ACM Transactions on Networking*, and served on their steering committee for 11 years. In 2006 he received the Koji Kobayashi Award for his work in computer communications. He has also been awarded the IEEE's Leonard G. Abraham Prize Paper Award in 1985 and 1987, for his papers on data and voice on CATV networks and the Manhattan Street networks, and the William R. Bennett Prize Paper Award in 1997, for his paper on an anonymous credit card.

HENNING SCHULZRINNE received degrees from Darmstadt University of Technology, Germany, the University of Cincinnati, and the University of Massachusetts in Amherst. He has held research positions at GMD Fokus, Berlin, and Bell Laboratories before joining the faculty of Columbia University, New York. He is currently chairing the Department of Computer Science. His research interests encompass real-time network services, ubiquitous and mobile computing, and network reliability. He is a co-author of more than 50 RFCs, including RTP, RTSP, SIP, and GIST.

STEPHEN SCHWAB is a senior principal investigator/scientist within SPARTA's security research division and holds an M.S. in computer science from Carnegie Mellon. His work spans all aspects of the research life cycle, from formulation of research ideas through conception, development, evaluation, and technology transition to production systems. His research focuses on experimental investigation and reduction to practical application of advanced research drawn broadly from the systems, networking, computer architecture, artificial intelligence, and information security communities. Recent research projects span the range from security infrastructure testbeds for accelerating near-term product development; rapid development of software technology to enable a new generation of high-speed, highly maintainable network packet analysis products; to long-term research and development efforts aiming to incorporate AI into network transport protocols and systems. Current activities include security architecture investigations for the NSF GENI effort, chief architect of a DARPA-funded National Cyber Range design effort, and co-PI of the DARPA-funded ZODIAC project.

BRUCE SIEGELL is a senior scientist in the Internet and Wireless Network Management Research Department at Telcordia. Most of his research activities at Telcordia have been in the areas of performance measurement and QoS for IP communication over various access and core network technologies, with the recent focus being on wireless technologies. He is currently involved in the DARPA SAPIENT and ZODIAC projects. He has also done research in fault management, and in parallel and distributed computing. He received his Sc.B. degree from Brown University in 1984 and his M.S. and Ph.D. degrees from Carnegie Mellon University in 1986 and 1995, respectively.

ANGELOS STAVROU [M] (astavrou@gmu.edu) is an assistant professor in the Computer Science Department and a member of the Center for Secure Information Systems at George Mason University, Fairfax, Virginia. He received his M.Sc. in electrical engineering, M.Phil., and Ph.D. (with distinction) in computer science all from Columbia University. He also holds an M.Sc. in theoretical computer science from the University of Athens, and a B.Sc. in physics with distinction from the University of Patras, Greece. His current research interests include security and reliability for distributed systems, security principles for virtualization, and anonymity with a focus on building and deploying large-scale systems. He is a member of the ACM and USENIX.

JONATHAN M. SMITH [F] is the Olga and Alberico Pompa Professor of Engineering and Applied Science and a professor of computer and information science at the University of Pennsylvania. He served as a program manager at DARPA, 2004–2006, and was awarded the OSD Medal for Exceptional Public Service in 2006. His current research interests range from programmable network infrastructures and cognitive radios to disinformation theory and architectures for computer augmented immune response.

There is an ongoing policy discussion as to whether providers should be allowed to block certain applications and protocols and lawsuits regarding whether one has rights to anonymity. ZODIAC is designed for those environments where security is valued more highly than anonymity.